Optična razpoznava notnih znakov s CRNN

Matic Isovski, Luka Šajn

Univerza v Ljubljani
Fakulteta za računalništvo in informatiko
E-pošta: mi6568@student.uni-lj.si, luka.sajn@fri.uni-lj.si

Abstract

Optical music recognition (OMR) is a field of research that investigates how to computationally read music notation. We study the use of Convolutional Recurrant Neural Network as an implementation of an end-to-end feature extractor and symbol sequence classifier. This approach is faster than the traditional and is not sensitive to image variability. We describe and demonstrate the use of PrIMuS dataset and supervised learning complexity. We also try to optimize model parameters with the Connectionist Temporal Classification loss function and back-propagation algorithms. At the end, we compare our minimalistic model to a popular app with integrated OMR technology.

1 Uvod

Kljub temu, da je tehnologija OMR (angl. Optical Music Recognition) v zadnjih 50. letih veliko napredovala, gre še vedno za zelo odprto problematiko. Problem je zelo kompleksen saj obstaja mnogo različic glasbenih notacij, ki bi jih moral celosten sistem prepoznati in se zato ponavadi gradijo modeli, ki so prilagojeni le na en spekter celotnega problema. Podpirajo npr. samo en tip zapisa (digitalen, sken, ročna pisava), eno različico pisave, določeno zvrst, itd. Razvila sta se dva pristopa reševanja tega problema: "deli in vladaj" [7] ter holistični [8]. Prvi razbije nalogo na več manjših in preprostejših nalog, katerih rešitve sestavijo končno. Najbolj osnovne naloge so razpoznava ter odstranitev notnega črtovja, določitev intonacije in taktovskega načina, klasifikacija notnih znakov ter lokalizacija notnih znakov. Klasifikacija simbolov je dosežena z algoritmi za primerjavo predlog. Osnovnejši pristop uporablja primerjavo predlog s površinskim pristopom, ki deluje neposredno na vrednostih slik in je močno občutljiv na spremenljivost predlog (afine transformacije), prav tako pa je potrebno predprocesiranje slik. Drugi pristop pa namesto neposredne primerjave uporablja ekstrakcijo in primerjavo značilk slike. Implementiran je s konvolucijsko globoko mrežo in je v primerjavi s prvim pristopom računsko znatno bolj optimalen ter ni občutljiv na spremenljivost predlog.

2 Pregled področja

En izmed bolj poznanih odprtokodnih programov za optično razpoznavo notnih znakov je Audiveris [1]. Poleg razpoznave notnih znakov omogoča tudi izvoz datoteke v MusicXML format za nadaljnjo uporabo v drugih programih. Razpoznava poteka v več fazah, v vsaki je program osredotočen na določen sklop (notni znaki, črtovje, besedilo, akordi, znaki za dinamiko itd.). Za vsakega od sklopov ima deskriptni razred, ki hrani seznam značilk, število razredov, uteži ter ostale parametre, ki se uporabljajo pri konfiguraciji nevronske mreže. Za vsak sklop se torej zgradi nov model (s shranjenimi že naučenimi parametri), na koncu pa se vse klasifikacije združi in prikaže, za kar poskrbi razred za upodabljanje (angl. render).

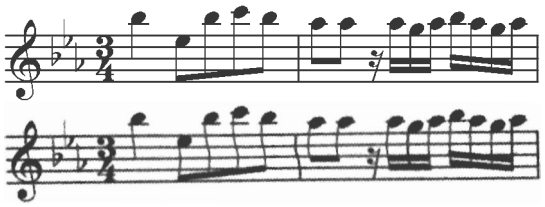
3 Zbirka notnih zapisov

Pri učenju modela smo uporabili zbirko notnih slik CameraPrIMuS, ki je razširjena različica zbirke PrIMuS [2] (angl. Printed Images of Music Staves), ki je nastala v sklopu raziskave na področju digitalnega zvoka ter slikovne obdelave. V članku [6] so predlagali in opisali uporabo CRNN (angl. Convolutional Recurrent Neural Network) modela za namene OMR. Model se je na omenjenih zbirkah učil s pomočjo povezovalno časovne klasifikacije (angl. Connectionist Temporal Classification, v nadaljevanju CTC), z algoritmom za vzvratno razširjanje (angl. backpropagation) [9] pa popravljal lastne parametre.

Zbirka vsebuje 87678 direktorijev, vsakega z dvema vhodnima vzorcema (navadna ter izkrivljena različica) ter 5 različnimi izhodnimi datotekami. Normalna vhodna slika je shranjena v PNG obliki, izkrivljena pa v JPG, vsebujeta pa izrezek vrstice z notnim črtovjem, natančneje glasbenih incipitom. Incipit [5] je zaporedje not, običajno prvih, uporabljenih kakor identifikacijska oznaka nekega glasbenega dela ali melodije. Vsebuje glasbeni ključ, predznake za določitev tonalitete, taktovski način ter od dva do pet taktov. Zaradi želje po izgradnji modela, ki bi dobro klasificiral note na slikah zajetih s pametnim telefonom, smo uporabili popačene različice vzorcev, saj vsebujejo podoben šum ter izkrivljenost. Primer obeh različic lahko vidimo na sliki (1).

Pri ocenjevanju modela si lahko pomagamo s petimi različnimi oblikami izhodnih datotek. Prva oblika je glas-

beno simbolična predstavitev izseka zakodiranega v MEI [3] (angl. Music Encoding Initiative). Naslednja oblika je PAE [4] (angl. Plaine & Easie Code), odprtokodni mednarodni knjižnični standard, ki prav tako omogoča vnašanje glasbenih incipitov. Direktorij vsebuje tudi dve poenostavljeni obliki glasbenega kodiranja: semantično ter agnostično. Semantično kodiranje vsebuje razredna imena vseh glasbenih znakov ter pripadajoče notne vrednosti tistim, ki jih je mogoče določiti. Agnostično kodiranje je podobno, le brez notnih vrednosti.



Slika 1: Primer navadne vhodne slike (zgoraj) ter popačene vhodne slike iz zbirke (spodaj).

4 Celostna globoka nevronska mreža

Zgradili smo celosten (angl. end-to-end) sistem, ki bo sam poskrbel za vse procesiranje potrebno za razpoznavo znakov. Imamo torej niz vhodnih slik, ki vsebujejo notno črtovje ter niz ustreznih zaporedij znakov. V namene iskanja značilk slike, po katerih bo sistem razpoznal znake, smo sestavili konvolucijsko nevronsko mrežo (angl. Convolutional Neural Network, krajše CNN). CNN se veliko uporablja za prepoznavanje slik in videa, sisteme priporočil, klasifikacijo slik, segmentacijo slik, medicinsko analizo slik, obdelavo naravnega jezika, ipd. Samo CNN pa ni kos nalogi, saj so note sekvenca znakov, pri čemer je za napovedovanje naslednjega znaka ključna informacija, kaj se nahaja pred njim. V te namene bomo uporabili rekurzivno nevronsko mrežo (angl. Recurrent Neural Netwrk, krajše RNN), saj zaradi svoje ciklične povezave omogoča delo s sekvencami.

Iz CNN in RNN smo sestavili konvolucijsko ponavljajočo se nevronsko mrežo (CRNN). To smo naredili tako, da smo izhod CNN povezali z vhodom v RNN. Pridobljene značilke torej pošljemo v RNN, ki pa je odgovorna za klasifikacijo znakov. Zahvaljujoč zbirki z glasbenimi incipti ter labelami lahko izvedemo nadzorovano učenje kar nad združenim modelom in ne nad vsakim blokom posebej. Pri tem igrajo glavno vlogo nizi s pravilnim zaporedjem notnih znakov v inciptih. Vhod, ki pride v RNN sloj si lahko predstavljamo kakor sliko, ki se nato razbije na stolpce, vsak stolpec pa vsebuje znak. RNN mora torej imeti izhod v velikosti za eno večje od števila razredov (v primeru kadar stolpec ne bo vseboval znaka). Ker nas segmentacija in lokalizacija razpoznanih znakov zaenkrat ne zanimata, lahko s pomočjo CTC metode izvajamo lokalno optimizacijo parametrov nevronske mreže za nazaj in tako bistveno pripomoremo k izboljšanju modela v fazi učenja.

4.1 Arhitektura

Arhitekturo modela lahko razberemo iz tabele 1. Vsi vhodni vzorci so v sivinski obliki različnih širin in višin ker je pa najmanjša višina vzorcev 128 slikovnih točk, smo toliko nastavili tudi velikost vhoda CNN. Vhodni del modela je zgrajen iz konvolucijskega bloka, ki vsebuje 3 konvolucijske sloje z inkrementalnimi velikostmi. Prvi zaznava 32 značilk, drugi 64, tretji pa 128. Vsak sloj črpa značilke v velikosti 3×3 in je povezan še v en sloj maksimalnega združevanja (angl. max-pooling) s filtri v velikosti 2×2, ki pomagajo pri optimizaciji mreže z nižanjem vzorčenja. V vseh konvolucijskih slojih se za aktivacijsko funkcijo uporablja ReLU (angl. Rectified Linear Unit), prav tako se pa vsakemu nizu vhodov (angl. batch) normalizira tako vrednost kot tudi dimenzije. Zadnji sloj konvolucijskega bloka se nato poveže z vhodom ponavljajočega se bloka, ki je zgrajen iz treh slojev. Prva dva sta sestavljena iz 256 BLST (angl. bidirectional long short-term memory) enot, ki iz filtrov poskušajo razpoznati niz notnih znakov. Tretji sloj je polno povezani (angl. fully-connected) sloj, ki ima 1782 nevronov (1781 razredov + 1 za "prazen" stolpec). Nad zadnjim slojem se izvede normalizirano eksponentno funkcijo (softmax), tako da imamo kot izhod verjetnosti pripadanja razredu.

Vhodna slika ($128 \times W \times 1$)
CNN blok
Conv2D ($32, 3 \times 3$), MaxPooling2D (2×2) Conv2D ($64, 3 \times 3$), MaxPooling2D (2×2) Conv2D ($128, 3 \times 3$), MaxPooling2D (2×2)
RNN blok
BLSTM (256) BLSTM (256) Dense (1782), softmax

Tabela 1: Zgradba CRNN modela.

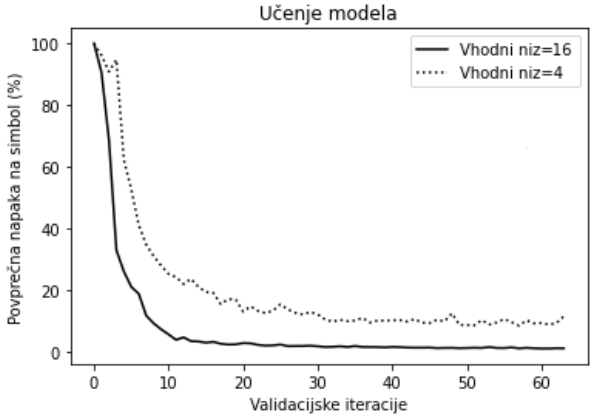
5 Učenje modela

Zbirko smo razdelili na učno (85%) in testno (15%). Učenje smo izvajali v iteracijah in v vsaki uporabili niz vhodov (angl. batch). Na vsakih 1000 iteracij smo izvedli še validacijo ter optimizacijo parametrov s pomočjo CTC funkcije.

5.1 Ocena

Primerjali smo uspešnost učenja z vhodnim nizom velikosti 4 slik, ter vhodnim nizom velikosti 16 slik (slika 2). Pri obeh vidimo na začetku velik padec v napaki, pri večjem nizu padec traja do nekje 6. iteracije, kjer pride do majhnega popravka in nato spet padanje, ki se pa po 12. iteraciji umiri. Pri učenju z manjšim nizom se popravek zgodi že takoj v 3. iteraciji potem pa strmo pada in se počasi po 20. iteraciji umiri. Jasno lahko vidimo, da velikost niza vpliva na uspešnost treniranja. Pri učenju z manjšim nizom pridemo do najboljše ocene z 8% napako, pri drugem pa do 1%, upoštevati je pa potrebno,

da je to povprečna napaka na simbol. V najslabšem primeru je torej napaka enaka $err = n \times err_s$, kjer je n število znakov v inciptu. Kljub temu, da z večjim vhodnim nizom dosežemo boljše rezultate, pa še vedno ni pametno kar večati števila slik. S prevelikim nizom namreč lahko pride do prekomernega prileganja, česar pa nočemo, saj je uporaba modela mišljena na različnih notnih zapisih (ki imajo sicer skupne lastnosti, ki bi jih CNN razpoznal). Pri učenju z nizom velikosti 16 slik smo do-



Slika 2: Graf povprečne napake za vsak simbol glede na iteracije.

segli povprečno simbolno napako (dalje PSBN) 1,1% oz. povprečno sekvenčno napako (dalje PSKN) 19,7%. Pri manjšem nizu pa smo dosegli dosti večjo napako - 7,7% PSBN ter 69,2% PSKN (tabela 2).

	$niz = 16$	$niz = 4$
PSBN (%)	1,1	7,6
PSKN (%)	19,7	69,2

Tabela 2: Povprečna sekvenčna ter simbolna napaka.

5.2 Analiza napake

V tabeli 3 lahko vidimo štiri najpogostejše razrede z napačno klasifikacijo ter njen delež glede na povprečno napako predstavljeno v tabeli 2. Najpogosteje se napačno

razred	delež (%)
barline	40,2
clef-C2	10,5
rest-quarter	1,9
gracenote-G4_eighth.	1,3

Tabela 3: Povprečna napaka glede na razred.

klasificira taktovska črta (v večini primerov jo zazna, kjer je ni), sledi ji glasbeni ključ (mezzo-soprano), nato pa še četrtinska pavza in okrasna osminka višine G4.

6 Testiranje modela

Model smo testirali na slikah brez šuma, ter na slikah zajetih s pametnim telefonom. Pri obeh sklopih smo uporabili 20 slik. Pri slikah brez šuma smo dosegli 100 % uspešnost. Slabše rezultate pa smo dosegli pri slikah zajetih s pametnim telefonom. PSBN je znašala 44% (tabela 4). Najpogosteje je bila napačno klasificirana taktovska črta (dodana), velikokrat je pa bil dodan tudi taktovski način. Pri napakah samih notnih znakov pa smo opazili, da se v 89% primerov zgodi napačna klasifikacija višine tona, v 7% primerov napačna klasifikacija dolžine tona, v 4% primerov pa napačna klasifikacija obojega. Opazili smo tudi, da se največ napak zgodi na primerih, ki ne vsebujejo močnih skupnih značilk zbirke oz. kadar vsebuje znake, ki so v zbirki prisotni redko ali sploh ne. Vsak incipit se začne z glasbenim ključem, ki mu sledijo predznaki, taktovski način ter nato note. V primerih, kadar slika ni vsebovala taktovskega načina, je namesto prve note klasificiralo enega od taktovskih načinov.

Rezultati testiranja programa Audiveris na zbirki slik PrIMuS so prav tako bili predstavljeni v članku [6] in sicer dosegel je 44.2 % PSBN. Direktne primerjave sicer ne moremo opraviti, saj je naš model bil učen in ocenjen na zbirki Corpus. Lahko pa predvidevamo, da se na omenjeni zbirki ne bi izkazal bistveno bolje, temveč kvečjemu slabše. Testirali pa smo ga na istih 40 slikah kakor naš model. Na slikah brez šuma se je izkazal enako kakor naš model. Na slikah zajetih s pametnim telefonom je dosegel nekoliko boljše rezultate kakor naš model (40,2% povprečne napake na simbol). Ker smo uporabili le 20 slik je malce nesmiselno primerjati točnost sistemov, lahko pa primerjamo njuno hitrost. Vidimo lahko, da je naš model veliko hitreje izvajal razpoznavo kakor Audiveris. Z večanjem števila testnih primerov bi zagotovo vplivali na oceno točnosti modelov, ne bi pa bistveno vplivali na primerjavo hitrosti. Lahko torej rečemo, da je uporaba celostne nevronske mreže bolj optimalen pristop, kakor uporaba več modelov za vsak sklop, kakor to počne Audiveris. Uporaba pristopa, ki smo ga izbrali mi, bi bila npr. smiselna pri mobilni aplikaciji.

	naš model	Audiveris
Simbolna napaka (%)	44	40,2
Čas izvajanja (s)	1,7	14

Tabela 4: Povprečna simbolna napaka in povprečen čas izvajanja.

Literatura

[1] Audiveris, Dosegljivo: https://github.com/Audiveris/audiveris. [Dostopano: 12.7.2012]

[2] PrIMuS dataset, Dosegljivo: https://grfia.dlsi.ua.es/primus/. [Dostopano: 9.7.2012]

[3] MEI, Dosegljivo: https://music-encoding.org. [Dostopano: 9.7.2012]

[4] PAE, Dosegljivo: https://en.wikipedia.org/wiki/Plaine_%-26_Easie_Code [Dostopano: 10.7.2012]

[5] Incipit, Dosegljivo: https://en.wikipedia.org/wiki/Incipit [Dostopano: 15.7.2012]

[6] Jorge Calvo-Zaragoza and David Rizo. End-to-end neural optical musicrecognition of monophonic scores.Applied Sciences, 8(4), 2018

[7] K. R. Varadarajan, "A divide-and-conquer algorithm for min-cost perfect matching in the plane,"Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280), 1998, pp. 320-329, doi: 10.1109/SFCS.1998.743466.

[8] Emanuele Carlini, Patrizio Dazzi, and Matteo Mordacchini. A ho-listic approach for high-level programming of next-generation data-intensive applications targeting distributed heterogeneous computingenvironment.Procedia Computer Science, 97:131–134, 2016. 2nd Inter-national Conference on Cloud Forward: From Distributed to CompleteComputing.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep feedforwardnetworks. InDeep Learning, chapter 6.5, pages 200–220. MIT Press, 2016.